

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Polonca Guček Frankovič

Zgodnje učenje programiranja

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Polonca Guček Frankovič

Zgodnje učenje programiranja

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn
Ljubljana, 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Študent naj razišče področje zgodnjega učenja programiranja ter primerja in v praksi preizkusi obstoječe programe, ki so primerni za osnovnošolce. Razišče naj, katere aktivnosti potekajo za popularizacijo zgodnjega učenja programiranja in naredi pregled spletnih strani, ki za učenje in utrjevanje učne snovi uporabljajo nove, privlačne interaktivne pristope.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Polonca Guček Frankovič, z vpisno številko **24010052**, sem avtor diplomskega dela z naslovom:

Zgodnje učenje programiranja (Early Learning of Programming)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Luke Šajna
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 31. avgusta 2014

Podpis avtorja:

Zahvaljujem se mentorju, doc. prof. Luki Šajnu, za pomoč in vzpodbudo ter viš. pred. dr. Borutu Batagelju za neizčrpen vir idej.

Zahvaljujem se možu in otrokom, ki so potrpežljivo ter z veliko naklonjenostjo in zanimanjem spremljali moje delo. Zahvaljujem se vsem prijateljem in znancem, ki so mi na pomoč priskočili s tehnično in programsko opremo, ali pa z mano modrovali ob kavicah. Posebno mesto si zasluži Boštjan, ki mi je dal začetni impulz.

Najbolj pa sem hvaležna staršema, ki sta mi nekoč davno omogočila pokukati v svet.

Kazalo

I.	Kazalo slik	
II.	Kazalo tabel	
III.	Povzetek	
IV.	Abstract	
1	Uvod	1
2	Slovenske spletne strani, namenjene učenju in utrjevanju šolske snovi	3
2.1	Dežela Lilibi	3
2.2	Moja matematika	4
2.3	Nauk.si	4
2.4	Portal izobraževalnih iger	4
2.5	Astra.si	5
3	Zgodnje učenje programiranja	7
3.1	Aktivnosti za popularizacijo zgodnjega učenja programiranja.....	7
3.1.1	CoderDojo	7
3.1.2	Code.org	7
3.1.3	Evropski teden programiranja	8
4	Pregled programov, namenjenih zgodnjemu učenju programiranja.....	9
4.1	Programi, primerni za osnovnošolce	9
4.1.1	Hopscotch	9
4.1.2	Move the turtle	10
4.1.3	Tynker.....	11
4.1.4	LEGO® Education WeDo™	13
4.1.5	Scratch	14
4.1.6	Light-bot, Cargo-bot in Kodable	15

4.2	Primerjava programov	16
5	Izvedba poletne šole programiranja.....	17
5.1	Cilji.....	17
5.2	Potek.....	17
5.3	Projekti otrok.....	18
5.4	Ugotovitve	19
6	Zaključek	21

Literatura

Priloga

I. Kazalo slik

Slika 2.1: Dežela Lilibi – primer naloge iz slovenščine	3
Slika 2.2: Moja matematika – primer naloge za 3. razred osnovne šole	4
Slika 4.1: Primer uporabe sposobnosti "Jump" v Hopscotchu	10
Slika 4.2: Move the turtle – primer vaje	11
Slika 4.3: Dragon journey v programu Tynker	12
Slika 4.4: Ukazni blok v Tynkerju	13
Slika 4.5: Primer programa v LEGO WeDo	14
Slika 4.6: Primer programa v Scratchu	15
Slika 5.1: Raziskovanje LEGO WeDo na poletni šoli programiranja	18
Slika 5.2: Projekti otrok, izdelani v poletni šoli programiranja in objavljeni na spletu	19

II. Kazalo tabel

Tabela 4.1: Primerjalna tabela programov <i>Hopscotch</i> , <i>Move the turtle</i> , <i>Tynker</i> , <i>LEGO WeDo</i> in <i>Scratch</i>	16
--	----

III. Povzetek

Cilj diplomskega dela je raziskati področje zgodnjega učenja programiranja, predvsem pa primerjati in v praksi preizkusiti obstoječe programe. V ta namen so bile raziskane aktivnosti za popularizacijo zgodnjega učenja programiranja. Narejen je bil pregled spletnih strani, ki za učenje in utrjevanje učne snovi uporabljajo nove, privlačne interaktivne pristope. Poudarjen je bil pomen zgodnjega učenja programiranja in narejena primerjava programov, ki so bili v ta namen razviti in so primerni za osnovnošolce. Programi so bili preizkušeni s skupino osnovnošolcev v poletni šoli programiranja. V njihovih zaključnih projektih je bilo na novo pridobljeno znanje programiranja povezano z interaktivnim učenjem matematike. Izkazalo se je, da med osnovnošolci vlada izjemno zanimanje za učenje programiranja in da so izbrani programi odličen pripomoček za doseg tega cilja.

Ključne besede: zgodnje učenje programiranja, osnovnošolci, kodiranje, algoritmično razmišljanje

IV. Abstract

The aim of the thesis is to explore the field of early learning of programming, particularly to compare and test the existing programmes in practice. To this purpose, activities for the popularisation of early learning of programming were explored. An overview of websites using new and attractive interactive approaches for knowledge acquisition and reinforcement was made. The importance of early learning of programming was emphasized and a comparison was made between the programmes developed for this purpose and suitable for basic school students. The programmes were tested with a group of basic school students attending a programming summer school. In their final projects, the newly acquired knowledge of programming was integrated with interactive learning of mathematics. It turned out that there is an extraordinary interest in learning of programming among the basic school students and that the selected programmes are an excellent tool for achieving this goal.

Key words: thesis, early learning of programming, coding, algorithmic thinking, basic school students

1 Uvod

Živimo v času, ko si življenja brez računalnika skoraj več ne predstavljamo. Z njegovo pomočjo komuniciramo, iščemo podatke in informacije, nakupujemo, plačujemo račune, beremo knjige in časopise ... Starši že predšolskim otrokom kupujejo tablične računalnike, da se namesto z igračami zamotijo z drsanjem po zaslonu. Vse bolj prisotni so tudi v šolah, kjer so učitelji redovalnice zamenjali z e-redovalnico, dnevnik z e-dnevnikom, uvedli spletne učilnice in začeli komunicirati s starši in učenci po e-pošti. Od učencev in dijakov se pričakuje, da seminarske naloge natipkajo v *Wordu*, govorne nastope pa opravijo ob *PowerPoint* predstavitvi, obdelujejo fotografije, zmontirajo kakšen film, snov poiščejo na internetu, domačo nalogo pa v spletni učilnici. Pri vsem tem je zanimivo to, da jih v šolah ničesar od tega ne učijo. V povprečni slovenski osnovni šoli učenec ne dobi niti ure računalništva, razen če ga izbere za izbirni predmet ali se vključi v krožek, česar pa vse šole ne ponujajo. Žal pa niti izbirni predmeti in krožki večinoma učencem ne dajo možnosti, da bi se poleg učenja uporabe računalnika srečali tudi s programiranjem. Problem je v prepočasnem odzivu šol na spremenjene razmere in pa v učiteljih; ti so zrasli in se izobraževali v drugačnem okolju, kot ga imajo njihovi učenci. Uporaba socialnih omrežij jim je bolj nuja kot želja, tablični računalniki predvsem nepotrební strošek, nenehno sledenje novim tehnologijam in možnostim, ki jih prinašajo, pa utrujajoče. Ne pozabimo na starše; tudi oni večinoma niso kos novim tehnologijam ...

Opisano stanje je bilo še nekaj let nazaj popolnoma sprejemljivo, a v času hitrega napredka tehnologij bi bilo prav, da mlajše generacije bolje pripravimo na življenje, ki jih čaka. Poznavanje osnovnih konceptov programiranja bo kmalu postalo nekaj vsakdanjega, nekaj, kar bo ponujalo boljše možnosti tistim, ki jim bodo domači. Prav bi bilo, da se tega začnemo zavedati in iskati načine, kako razbiti mite o programiranju kot nečem zahtevnem in dolgočasnem. Ključno vlogo pri tem bodo v prihodnosti morale odigrati šole, ki pa potrebujejo ustrezne kadre. Šele ko bodo učitelji razumeli, da je vse, kar potrebujejo, le nekaj poguma in dobre volje, lahko pričakujemo premike na bolje.

V diplomski nalogi se zato lotevamo področja programiranja za otroke. V začetni fazi bomo opravili pregled slovenskih spletnih strani, na katerih so uporabljeni modernejši pristopi k podajanju in utrjevanju učne snovi, in to uporabili za izhodišče pri praktičnem delu. Osrednji

del naloge bo namenjen raziskovanju, kaj in kako je narejenega za spodbujanje zgodnjega učenja programiranja in kakšna je ponudba programov, ki pri otrocih spodbujajo algoritmično razmišljanje ali pa so namenjeni zgodnjemu učenju programiranja. Temu bo sledil praktični del, to je izpeljava mini poletne šole računalništva, na kateri bomo z otroki preizkusili več izbranih programov in v programu *Scratch* izdelali samostojne projekte. Projekti bodo imeli skupen cilj: interaktivna izobraževalna igra s področja matematike.

Cilj diplomske naloge je izbrati, primerjati in v praksi preveriti nekaj programov, ki učijo otroke programirati ter ugotoviti, kako se v praksi otroci spopadajo z izzivi programiranja, in ali za programiranje izkazujejo interes na nivoju kot ga pričakujemo.

2 Slovenske spletne strani, namenjene učenju in utrjevanju šolske snovi

Računalnike uporabljajo vedno mlajši otroci, ki jih bolj kot knjige privlačijo računalniške igrice. Tudi zato so klasične načine poučevanja in učenja začeli dopolnjevati novi pristopi. Zanimalo nas je, kaj drugačnega in hkrati kvalitetnega lahko slovenski otroci, majhni in malo večji, najdejo za utrjevanje šolske snovi na spletnih straneh in kaj imajo na voljo učitelji, ki želijo pouk občasno popestriti z uro v računalniški učilnici.

2.1 Dežela Lilibi

Dežela Lilibi [1] je interaktivni izobraževalni portal, ki je nastal kot interaktivno gradivo k učbenišskemu kompletu Lili in Bine. Vsebuje didaktične igrice, poučne interaktivne naloge z zvočnimi navodili, pesmice, animacije, izobraževalne filme ter učne liste za vse predmete iz predmetnika v prvem triletju osnovne šole (Slika 2.1). Učiteljem ponuja tudi dodatna gradiva in priprave, ki jih lahko uporabijo pri pouku.



Slika 2.1: Dežela Lilibi – primer naloge iz slovenščine

2.2 Moja matematika

Moja matematika [2] je spletna vadnica, namenjena utrjevanju znanja matematike učencev od 1. do 5. razreda osnovne šole (Slika 2.2). Primerna je za uporabo doma ali v šoli. Ponuja privlačne interaktivne naloge, izdelan pa ima tudi sistem nagrajevanja s čebelicami, mesečnimi medaljami in priznanji. Učiteljem omogoča uporabo v razredu, pripravo spletnih domačih nalog in vpogled v doseganje učnih ciljev posameznih učencev.



Slika 2.2: Moja matematika – primer naloge za 3. razred osnovne šole

2.3 Nauk.si

Spletna stran *Nauk.si* [3] vsebuje gradiva s področja matematike, fizike, logike in računalništva za osnovne in srednje šole. Gradiva so sestavljena iz uvodne teorije, ki ji sledijo interaktivne naloge. Med drugim so uporabljena gradiva s spletnih strani *E-um* [4] (interaktivna gradiva iz matematike za osnovno in srednjo šolo), *Aktivna matematika* [5] in *E-va* [6] (interaktivna gradiva iz fizike za osnovno in srednjo šolo).

2.4 Portal izobraževalnih iger

Portal izobraževalnih iger [7] vsebuje interaktivna učna gradiva s področja računalništva, ki so nastala kot samostojni projekti študentov Katedre za računalništvo Pedagoške fakultete Univerze v Ljubljani.

2.5 Astra.si

Spletna stran *Astra.si* [8] je delo altruističnega mladeniča, namenjena predvsem srednješolcem in študentom, ki želijo z ogledom video posnetkov osvežiti ali poglobiti svoje znanje matematike.

3 Zgodnje učenje programiranja

Živimo v svetu, ki mu vlada programska oprema. V prihodnosti bo nepoznavanje računalniškega jezika podoben izziv kot je danes nepismenost. Kljub temu, da tudi v prihodnosti ne bo vsako delo vključevalo programiranja, je ključnega pomena, da se otroci naučijo kodirati. Računalniško razmišljanje kombinira matematiko, logiko in algoritme in nas uči novih načinov razmišljanja. Uči nas, kako velike probleme razdrobiti v zaporedje manjših, bolj obvladljivih, kar nam omogoča reševanje kompleksnih problemov na učinkovite načine, ki delujejo tudi v velikem obsegu. Pomaga nam pri pretvorbi rešitev specifičnih problemov v rešitve bolj splošne narave. [9]

Računalniško razmišljanje je veščina, ki bi se je morali vsi naučiti. Pomagalo bo razumeti in obvladati tehnologije vseh vrst in reševati probleme na skoraj vseh področjih.

3.1 Aktivnosti za popularizacijo zgodnjega učenja programiranja

Aktivnosti za popularizacijo zgodnjega učenja programiranja (sploh v Sloveniji) ni zelo veliko, večinoma so v sklopu promocije programiranja na splošno. Nekaj najbolj opaznih je naštetih v nadaljevanju.

3.1.1 CoderDojo

CoderDojo je internacionalno neprofitno gibanje, odprta znamka Fundacije CoderDojo, predano računalniškemu izobraževanju otrok. [10]

Nastalo je leta 2011, do danes pa se je razširilo v 48 držav, tudi v Slovenijo. Njegovo poslanstvo je širjenje računalniške znanosti in pismenosti med mladimi na nekoliko drugačen način, kot to zahteva formalno izobraževanje. Tečaji programiranja so brezplačni, vsi sodelujoči v gibanju pa so prostovoljci. [11]

3.1.2 Code.org

Code.org [12] je ameriška neprofitna organizacija, katere cilj je omogočiti izobraževanje s področja računalništva v vseh šolah čim večjemu številu otrok, ne glede na spol, barvo in

poreklo. V okviru kampanje Hour of code ponuja privlačne brezplačne tečaje osnov programiranja v različnih programskih jezikih.

3.1.3 Evropski teden programiranja

Evropska komisija spodbuja učenje programiranja. V ta namen bodo potekale delavnice po Evropi, tudi v Sloveniji, in na spletu, pripravili pa so tudi brezplačna gradiva v slovenščini. Delavnice bodo uvod v Evropski teden programiranja, ki bo potekal med 11. in 17. oktobrom 2014. Namen pobude Evropske komisije je narediti programiranje bolj prepoznavno, zato si želijo na dogodkih in v učilnicah zbrati čim več otrok, staršev, učiteljev, podjetnikov. [13]

4 Pregled programov, namenjenih zgodnjemu učenju programiranja

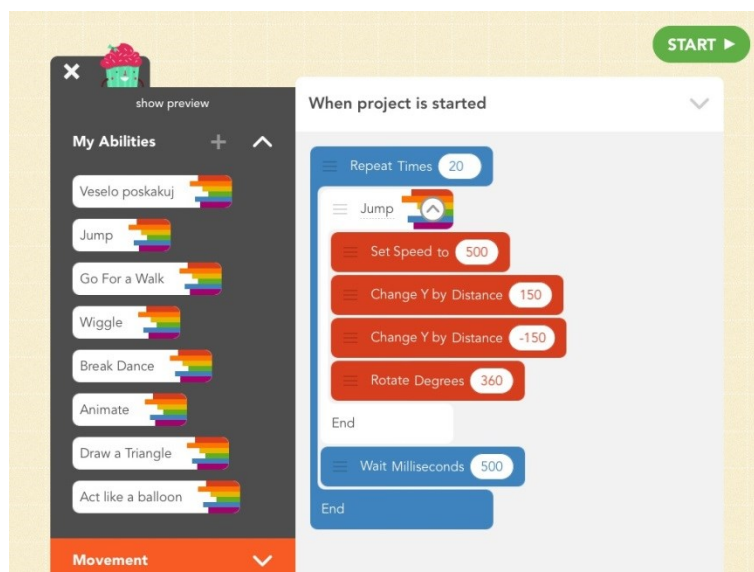
Na trgu je že veliko aplikacij, ki z različnimi pristopi skušajo približati programiranje vedno mlajšim otrokom. Narejene so za različne platforme, učenja programiranja pa se lotevajo iz različnih zornih kotov. Na tem mestu bomo omenili le tiste, med katerimi smo izbirali najprimernejše za delo z osnovnošolci, ki se s programiranjem srečujejo prvič: *Daisy the Dino* [14], *Hopscotch* [15], *DynamicArt* [16], *Kodu* [17], *Scratch* [18], *Alice* [19], *Move the Turtle* [20], *AppInventor* [21], *Kojo* [22], *Greenfoot* [23], *Tynker* [24], *LEGO® Education WeDo™* [25], *LEGO® Mindstorms®* [26], *Light-bot™* [27], *Cargo-bot* [28], *Kodable* [29].

4.1 Programi, primerni za osnovnošolce

Za primerjavo smo izbrali nekaj aplikacij, ki so se nam zdele najbolj primerne za prvi stik slovenskih osnovnošolcev s programiranjem. Glavna kriterija pri izbiri sta bila privlačna in jasna komunikacija aplikacije z uporabnikom ter primeren nivo zahtevnosti. V mislih pa smo imeli tudi primernost in prilagojenost aplikacije za uporabo pri pouku v osnovnih šolah.

4.1.1 Hopscotch

Hopscotch [15] je aplikacija za iPad, ki omogoča otrokom, da na način potegni/spusti sestavljajo bloke kode in s tem kreirajo svoje lastne programe. Namenjena je učenju programiranja brez tipkanja, prednost pa daje gradnji aplikacije pred mučenjem s sintakso. Ima nekaj že vnaprej pripravljenih blokov ukazov (npr. "nariši spiralo", "pleši", "obnašaj se kot balon" ...), iz katerih se lahko otrok uči, jih spreminja in popravlja. Otroku omogoča, da tudi sam gradi nove bloke in jih shrani za kasnejšo uporabo. Tako pripravljene funkcije ali podprograme *Hopscotch* imenuje sposobnosti (Abilities). Ko uporabnik sposobnost združi z dogodkom (npr. ko se iPad strese, ko je zaznan glasen hrup ...), ustvari pravilo (Rule). Tako lahko npr. ustvarimo pravilo, da naj se izbrana figura obnaša kot balon vsakič, ko iPad zazna glasen hrup, ali pa npr. nekajkrat poskoči, ko zaženemo projekt (Slika 4.1).



Slika 4.1: Primer uporabe sposobnosti "Jump" v *Hopscotchu*

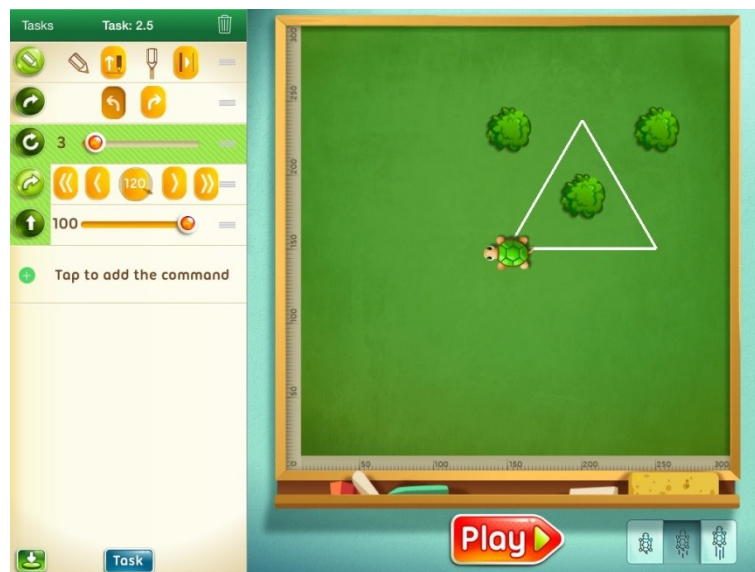
Hopscotch omogoča uporabnikom, da svoje projekte objavljajo in predelujejo projekte, ki so jih objavili drugi. Namesto vsesplošno popularnega všečkanja so ustvarjalci uporabili drug pristop: izpostavili so projekte, ki so bili največkrat predelani (most branched).

Aplikacija trenutno še ne vsebuje pogojnih stavkov, logičnih operatorjev, ne omogoča izdelave lastnih figur in uporabe zvoka. Ima le preprosto repeat zanko in ukaz wait. Ker je v fazi intenzivnega razvoja, lahko kmalu pričakujemo bolj pestro ponudbo ukazov.

4.1.2 Move the turtle

Move the turtle [20] je aplikacija za iPad in iPhone, ki za učenje osnov računalniškega programiranja uporablja implementacijo Logo programskega jezika. Naloga otroka je, da z ukazi premika želvo, ki med premikanjem pušča sled, kar lahko izkoristimo za izris različnih oblik.

Zaslon je razdeljen na dva dela: leva stran je rezervirana za menije in gradnjo programa, na desni je igralna/risalna površina z želvo na sredini. V bližini želve je tudi moder diamant, do katerega morajo otroci v enem ali več korakih pripeljati želvo, pri tem pa izpolniti še druge specifične naloge (Slika 4.2).

Slika 4.2: *Move the turtle* – primer vaje

Želvo otroci premikajo s sestavljanjem navodil, ki so večinoma preprosta (premakni, obrni, izberi barvo, zaigraj ton ...) in jih lahko prilagajajo. Npr.: premakni se 100 korakov proti severu.

Otroci so v svet programiranja vodeni skozi tri poglavja, prilagojena različnim starostnim skupinam, glede na matematične koncepte, ki jih poglavja uporabljajo (npr. mlajši še niso seznanjeni z geometrijo). Vsako poglavje vsebuje devet stopenj. Naloge imajo kratka in jasna navodila ter rešitve. Za napredovanje na višjo stopnjo je potrebno nalogo vsaj delno rešiti.

Move the turtle otroke seznani z zankami, procedurami, spremenljivkami in pogojnimi stavki. Otrokom omogoča graditi samostojne procedure, ki jih lahko shranijo in kasneje uporabijo v drugih projektih.

Aplikacija vsebuje tudi zelo uporabno knjižnico že pripravljenih primerov, na katerih se lahko otroci učijo, jih spreminjajo, ali pa uporabijo v svojih projektih. Podpira več profilov, kar omogoča, da na eni napravi spremljamo napredek več posameznikov.

4.1.3 Tynker

Tynker [24] je plačljiva aplikacija, ki deluje na sistemih Windows, Android in iOS (iPad).

Tynker vsebuje serijo vaj, s katerimi se otroci učijo konceptov programiranja. Probleme rešujejo s sestavljanjkami: koščki sestavljanke so ukazi, ki jih je potrebno pravilno sestaviti v

ukazne bloke. Zahtevnost vaj se postopoma stopnjuje, prav tako se postopoma dodajajo novi ukazi in kombinacije.

Na iPadu so vaje organizirane v tematsko obarvane avanture:

- CODEY'S QUEST – z uporabo logike in zank peljemo Codeya domov
- DRAGON JOURNEY – funkcije in procedure pomagajo zmaju pri potovanju skozi gozd (Slika 4.3)
- LAZER RACER – z dirkalnikom rišemo kompleksne geometrijske oblike - prostorska vizualizacija (gibanje po koordinatnem sistemu)
- LOST IN SPACE – pogojna logika pomaga astronautu do njegove baze na Luni

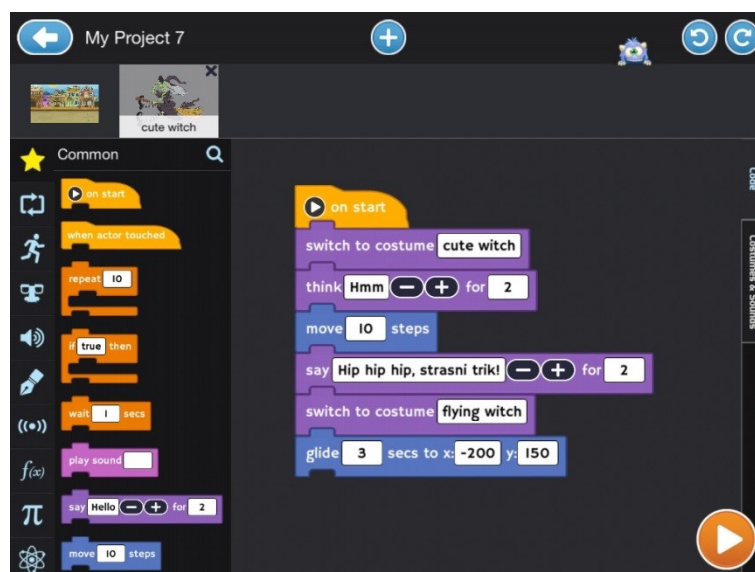


Slika 4.3: Dragon journey v programu *Tynker*

Odlična aplikacija s privlačno in prijazno grafiko pa ima tudi nekaj slabosti. Nekatere vaje imajo nejasna in nepopolna navodila. Vaje so zaklenjene, zato je napredovanje v vsako naslednjo vajo mogoče le ob predhodno uspešno opravljeni prejšnji vaji. Največjo slabost pa vidimo v tem, da se velikokrat pravilna rešitev ne prepozna kot pravilna, ker se ne ujema z rešitvijo, ki so jo predvideli razvijalci aplikacije.

Otroci pa se imajo možnost preizkusiti tudi v BUILD GAMES, kjer gradijo svoje igre in aplikacije (Slika 4.4). Pri tem uporabljajo isti vizualni programski jezik kot v vajah. Pripravljenih je več sto likov in ozadij, razdeljenih v 5 tematskih skupin, pa tudi 9 žanrsko

različnih iger, ki služijo kot predloge. V projekte lahko vključujejo zvoke in fizikalni pogon (physics engine), določajo svoje funkcije in spremenljivke.



Slika 4.4: Ukazni blok v Tynkerju

Podobno so pripravljene vaje tudi v okolju Windows. V okviru iniciative "Hour of code" je del Tynkerja v tem okolju prosto dostopen, za več pa je potrebno plačilo. Za razliko od aplikacije na iPadu tu sam izbiraš, v kakšnem vrstnem redu boš reševal vaje.

Tynker je programski jezik, ki ga za učenje programiranja v tujini (po njihovih podatkih) uporablja že več tisoč šol. Poskrbeli so za obsežna gradiva za učitelje in spletno učilnico. Žal pa za naše razmere predstavlja kar hud finančni zalogaj.

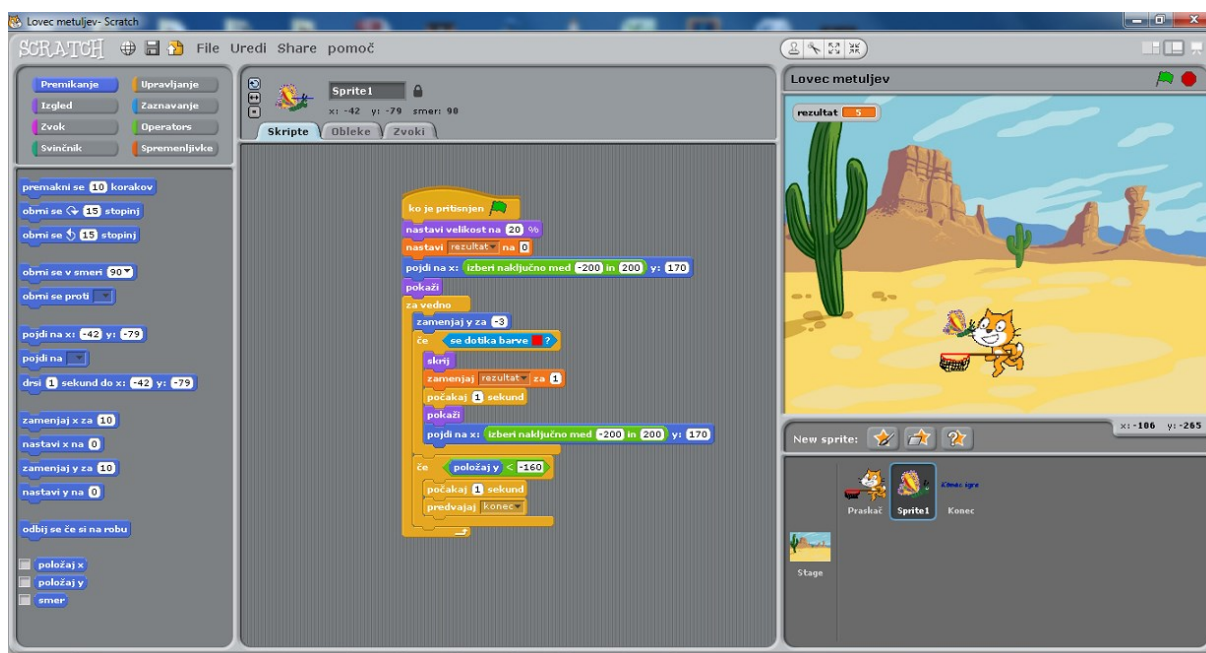
4.1.4 LEGO® Education WeDo™

Osnovni komplet LEGO® Education WeDo™ [25] (v nadaljevanju LEGO WeDo) sestavlja več kot 150 skrbno izbranih elementov, vključno z motorjem, senzorjem gibanja, senzorjem nagiba in USB priključkom. V kombinaciji s pripadajočo programsko opremo (LEGO Education WeDo Software) omogoča otrokom graditi in programirati preproste LEGO modele, ki so priključeni na računalnik.

"Povleci in spusti" programska oprema temelji na ikonah (ikona za senzor gibanja, zvok, vklop motorja ...), ki jih sestavljamo v ukazne bloke (Slika 4.5). Vsebuje tudi navodila za izdelavo 12 modelov in primere programov zanje.

Scratch je vizualno zelo privlačen, z možnostjo ustvarjanja lastnih figur, ozadij in zvokov pa omogoča neizmerno kreativnost. Za nas pomembna prednost *Scratcha* je tudi v slovenščino preveden uporabniški vmesnik in pa knjiga [30] v slovenščini.

Svoje projekte lahko otroci delijo na spletu, zaradi velike popularnosti *Scratcha* pa tam zase najdejo vse, kar potrebujejo: ideje, rešitve, vodiče, ...



Slika 4.6: Primer programa v *Scratchu*

Scratch pa ima še eno izjemno kvaliteto. Z *LEGO WeDo* USB priključkom ga lahko namreč med drugim povežemo z *LEGO* modeli in tako otrokom omogočimo še vpogled v svet robotov. In za razliko od cenovno zelo neugodne *LEGO WeDo* programske opreme je *Scratch* prosto dostopen vsem. Z nekaj preprostimi ukazi lahko otroci svoje *LEGO* modele povežejo z dogajanjem na zaslonu in se preselijo v 3D svet ...

4.1.6 *Light-bot, Cargo-bot in Kodable*

Light-bot [27], *Cargo-bot* [28] in *Kodable* [29] so primeri aplikacij, kjer je za učenje osnovnih konceptov programiranja uporabljenih le nekaj osnovnih ukazov, s katerimi premikamo robotka, dvigalo ali skuštrano kepico. Otroke učijo pomembnosti zaporedja, obravnavajo zanke in pogojne stavke, vpeljujejo procedure, težavnost nalog pa se stopnjuje.

4.2 Primerjava programov

Na prvi pogled zelo različne aplikacije imajo nekaj pomembnih skupnih točk. Vse so izredno privlačne za oko, enostavne za razumevanje in uporabo ter primerne za uporabo v šolah. Večina ima že pripravljena dodatna gradiva za učitelje (Tabela 4.1).

	<i>Hopscotch</i>	<i>Move the turtle</i>	<i>Tynker</i>	<i>LEGO WeDo</i>	<i>Scratch</i>
prosto dostopen	✓	×	delno	×	✓
okolje	iOS (iPad)	iOS (iPhone, iPad, iPod touch)	Web, Android, iOS	PC, MAC	Windows, OS X, Ubuntu
gradiva za učitelje	✓	×	✓	✓	✓
gradniki	ploščice	vrstica (ikona, drsnik)	koščki sestavljanke	ikone	koščki sestavljanke
vaje	×	✓	✓	×	×
zanke	✓	✓	✓	✓	✓
koordinatni sistem	✓	✓	✓	×	✓
spremenljivke	✓	✓	✓	×	✓
pogojni stavki	×	✓	✓	×	✓
procedure	✓	✓	✓	×	✓
matematične operacije	×	✓	✓	osnovne	✓
naključnost	✓	✓	✓	✓	✓
gnezdenje	✓	✓	✓	×	✓
logični operatorji	×	✓	✓	×	✓

Tabela 4.1: Primerjalna tabela programov *Hopscotch*, *Move the turtle*, *Tynker*, *LEGO WeDo* in *Scratch*

5 Izvedba poletne šole programiranja

5.1 Cilji

Zanimalo nas je, kako se prej omenjeni programi obnesejo v praksi. Naš osnovni cilj je bil, da otroci do konca poletne šole preizkusijo čim več različnih programov in v *Scratchu* izdelajo uporabne interaktivne naloge (igre) s področja matematike. Ker smo želeli čim bolj različne in kreativne rešitve, so dobili le eno navodilo: naj bo projekt matematično obarvan, pri čemer naj se zadržujejo znotraj preprostih okvirjev štetja, barv in osnovnih računskih operacij.

Prav tako smo želeli doseči, da bi otroci razumeli pomen medsebojnega sodelovanja in izmenjave informacij ter t.i. brainstorminga, kar skupaj vodi do boljših rešitev v krajšem času.

5.2 Potek

Organizirali smo petdnevno, 20-urno mini poletno šolo programiranja. S skupino osmih otrok v starosti od 9 do 13 let smo preizkusili *Light-bot*, *Cargo-bot*, *Kodable*, *LEGO WeDo* (Slika 5.1) in *Scratch*. Igre *Light-bot*, *Cargo-bot* in *Kodable* smo uporabljali za ogrevanje, večino časa pa smo posvetili *Scratchu*. Pri delu smo poudarjali pomen sodelovanja, pomoči, izmenjave informacij in timskega dela. Izpostavljali smo težave, na katere so pri programiranju naleteli posamezniki, in jih skupaj odpravljali. Otroci so drug drugemu pomagali pri testiranju projektov ter ponujali nove ideje in rešitve. Prve tri dni smo intenzivno spoznavali *Scratch*, v nadaljevanju pa so imeli otroci čas za izdelavo svojih projektov. Pri učenju *Scratcha* nam je bila v pomoč knjiga *Scratch* [30], avtorice Sonje Lajovic. Ukaze smo spoznavali po skupinah, po nekaj najbolj osnovnih iz vsake skupine, v nadaljevanju pa smo se učili na primerih, ki smo jih gradili skupaj. Otrokom smo predstavili koordinatni sistem, saj so se z njim srečali prvič.

Nekaj ur smo namenili *LEGO* kockam, programu *LEGO WeDo* ter povezavi *LEGO* in *Scratcha*. Sestavljanje modelov po navodilih je izziv ene vrste, doseči, da bo model deloval, pa druge. Preizkusili smo nekaj modelov, zmagal je krokodil, ki zazna pred svojimi velikimi čeljustmi gibanje, zato usta zapre in glasno hrusta. Pojedel je kar nekaj papirnatih kroglic in lego kock ter ugriznil nekaj prstkov.



Slika 5.1: Raziskovanje *LEGO WeDo* na poletni šoli programiranja

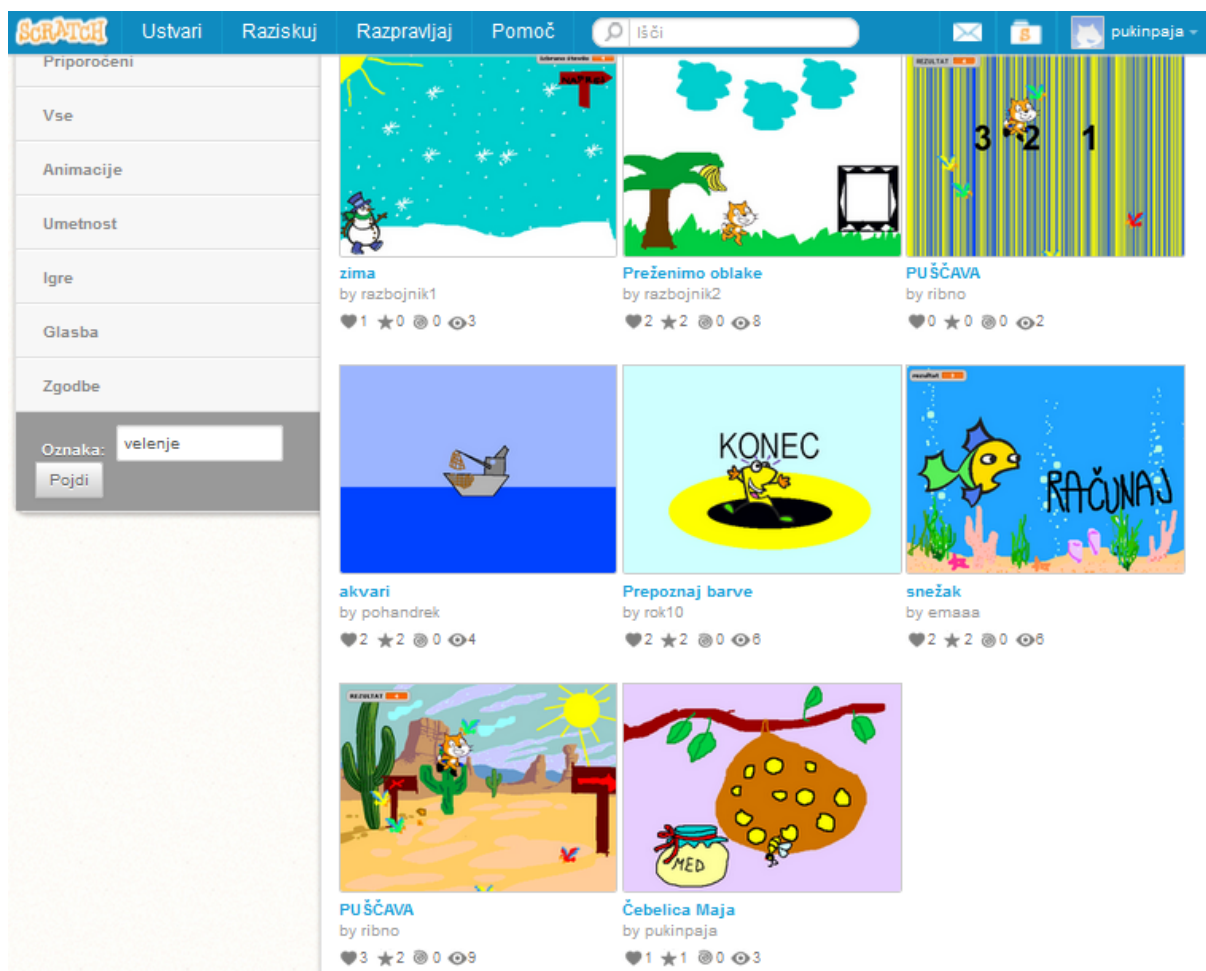
Kocke *LEGO* pa smo povezali tudi s *Scratchem*. S skupnimi močmi smo napisali program, ki je povezal dogajanje na zaslonu s fizično akcijo. Ko se je mucek na zaslonu dotaknil žoge, je velika *LEGO* noga brcnila. Spreminjali smo moč brce in pripravljena žogica je v veliko veselje otrok letela po učilnici. Otroci so predlagali, da bi nogi dodali senzor gibanja. Nastal je program, ki je združil dve akciji: ko je senzor zaznal gibanje, sta istočasno brcnila noga iz kock in mucek na ekranu.

Končane projekte v *Scratchu* smo zadnji dan še objavili in preizkusili v spletu.

Otroci so zadnji dan preizkusili tudi program *Hopscotch*. Programa *Tynker* in *Move the Turtle* pa smo zaradi pomanjkanja časa v manjši skupini preizkusili kasneje.

5.3 Projekti otrok

V sklopu poletne šole so otroci izdelali nekaj samostojnih projektov v programu *Scratch* (Slika 5.2). Namenjeni so utrjevanju matematike na nivoju nižjih razredov osnovne šole. Področja, ki so se jih lotili, so poznavanje barv, štetje, seštevanje in odštevanje, poštevanje. Otroci so bili pri izdelavi projektov zelo kreativni, visoko motivirani in večinoma samostojni. Projektov so se lotili z vso resnostjo in z jasno željo po dobri vsebini. Uporabili so vse znanje, ki so ga v samo nekaj dneh prejeli, do izraza pa so prišli njihovi interesi; nekateri so večino časa posvetili likovnemu izgledu izdelka, drugi so se raje ukvarjali s kompleksnejšo vsebino. Opis končnih projektov je v Prilogah.



Slika 5.2: Projekti otrok, izdelani v poletni šoli programiranja in objavljeni na spletu

5.4 Ugotovitve

Odziv otrok je bil fantastičen. Zavzeto so garali od prve do zadnje minute in v zelo kratkem času so nastali uporabni izdelki. Imeli so odlične ideje, vztrajno so iskali rešitve in pri tem pokazali veliko mero iznajdljivosti s prepisovanjem in predelavo kode iz projektov, na katerih smo se učili (jih naredili skupaj) ali tistih, ki so jih našli na spletu. Največ težav so imeli pri pravilni izbiri zank, čeprav s samim razumevanjem zank niso imeli problemov.

Aplikacija *Kodable* je bila zanimiva le kratek čas, opisali so jo kot preveč enostavno. Tudi *Cargo-bot* ni požel kaj več od začetnega navdušenja, a iz drugega razloga: je prezahteven. Je pa *Light-bot* opravil veliko delo. Otroci so bili primorani začeti algoritmično razmišljati, iskati in odpravljati svoje napake, iskati drugačne rešitve in vztrajati. Ko so med seboj primerjali rešitve, so odkrili, da lahko isti problem rešijo na različne načine. Igra je med otroki sprožila

tekmovalnost in jih še dodatno motivirala. Pri igranju *Light-bota* so se prvič srečali z zankami in procedurami, in jih tudi brez večjih težav usvojili.

LEGO WeDo je bil izredno privlačen zaradi kock, ki jim je program vdihnil življenje. Ker so osnovne principe programiranja in sestavljanja ukazov na sicer malo drugačen način spoznali že pri *Scratchu*, jim programiranje v *LEGO WeDo* ni predstavljalo težave. Nastavljali so moč in smer vrtenja motorja, dodajali zvok, pogoje, vezane na senzorje ... in uporabljali preproste zanke.

Ko smo *LEGO* model povezali s *Scratchem* in napisali program, ki je dogajanje na ekranu povezal z našim modelom (brcanje žoge), je bilo to zaradi preprostosti in zabavnega rezultata odlična motivacija za nadaljnje raziskovanje.

Hopscotch smo na iPadu otrokom pokazali zadnji dan. Po izkušnji s *Scratchem* se jim je zdel otročji in preveč omejujoč. Strinjali smo se, da jim ne predstavlja več izziva. Kasneje smo to trditev preverili še na skupini, ki izkušnje s *Scratchem* še ni imela. Tudi otrokom v tej skupini je po kakšni uri zanimanje za aplikacijo splahnelo.

Kasneje smo s skupino štirih otrok na iPadu preizkusili še *Move the turtle* in *Tynker*. Malo večji izziv je predstavljala aplikacija *Move the turtle*, predvsem zaradi pomanjkanja specifičnih znanj iz matematike, medtem ko *Tynkerjeve* vaje niso delale težav. Otroci so se preizkusili tudi v izdelavi lastnega projekta v *Tynkerju*, kjer jih je navdušila predvsem grafika, programiranje pa je zelo podobno tistemu v *Scratchu*. Največja ovira je bila angleščina.

Otroci so se vse dni problemov lotevali zelo sproščeno. Za spoznavanje, razumevanje in uporabo koordinatnega sistema je bilo nekaj minut teorije dovolj. Brez težav so se znašli v vseh programskih okoljih. Zelo hitro so sprejeli spremenljivke in jih skoraj vsi tudi uporabili v svojih projektih. Največ časa smo porabili za zanke, saj so otroci do prave izbire prišli bolj s poskušanjem kot z resnim razmislekom.

6 Zaključek

V diplomski nalogi smo raziskovali dogajanje in možnosti na področju zgodnjega učenja programiranja. Seznanili smo se z aktivnostmi, ki v ta namen potekajo v Sloveniji in tujini. Pregledali smo široko paleto v ta namen razvitih programov in naredili primerjavo nekaj izbranih, ki so primerni za učenje osnovnošolcev. Programe smo preizkusili s skupino otrok v poletni šoli programiranja in jih usmerili v izdelavo interaktivnih iger s področja matematike. Interaktivni in inovativni pristopi k učenju in utrjevanju učne snovi postajajo namreč vedno bolj priljubljeni, nekaj primerov smo opisali tudi v tem diplomskem delu.

V Sloveniji še vedno velja splošno prepričanje, da se programirati začne tam nekje v srednji šoli, da je programiranje zahtevno in dolgočasno, ter rezervirano za posebneže. Šole za to področje ne kažejo zanimanja, predvsem zaradi večinoma programiranja neveščih in slabo informiranih učiteljev, ki so obremenjeni s predsodkom, da tega ne zmorejo. Mlajše generacije na področju uporabe računalnikov prehitevajo starejše, ki pa obvladujejo področje izobraževanja. Ker otroci večino dneva preživijo v šoli, bi bilo šolsko okolje idealno za vodeno spoznavanje različnih možnosti uporabe računalnika, tudi programiranja.

V diplomskem delu smo želeli opozoriti na široko izbiro privlačnih in preprostih pristopov k učenju osnov programiranja in algoritmičnega razmišljanja. Morda opisana in v praksi temeljito preverjena privlačnost, preprostost in dostopnost aplikacij na različnih napravah opogumi odrasle, da se s svojimi otroki ali učenci lotijo novih izzivov. Res je, da za večino tovrstnih aplikacij potrebujemo iPad, ki ni dostopen vsem; prav tako velik del aplikacij zahteva (simbolično) plačilo. Vseeno pa bi lahko ti programi predstavljali sprejemljiv kompromis v času, ko se oblikujeta dva tabora: v enem odločno nasprotujejo pretirani rabi računalnikov pri otrocih, v drugem pa spodbujajo k uporabi že malčke v pleničkah.

Programe smo z izvedbo poletne šole programiranja temeljito preizkusili z otroki, starimi med 9 in 13 let. Učenje je potekalo v sproščenem, kreativnem vzdušju, kjer otroci niso niti opazili, kako zelo razmišljajo. Navdušeni so bili nad vsemi ponujenimi programi. Presenetila nas je lahkota, s katero so informacije prenašali v prakso in iznajdljivost pri prepisovanju in predelavi kode. Delali so samostojno ali v parih, ves čas pa so tudi opazovali in občudovali druge, dajali ideje in pomagali iskati rešitve. Aktivno so raziskovali zmožnosti programov in izredno hitro napredovali. Zanke, spremenljivke, pogojni stavki, logični operatorji ... v manjšem obsegu niso predstavljali težav. Zapletalo se je pri kompleksnejših rešitvah, pri

gnezdenju in na splošno pri doslednosti. Presenetil jih je predvsem pomen pravilnega vrstnega reda ukazov.

Na nepričakovano težavo smo naleteli, ko smo želeli delati s *Hopscotch*em. Izkazalo se je, da je večini otrok iPad nedosegljiv, zato smo bili prisiljeni improvizirati z manjšim številom izposojenih. Poskusili smo si jih izposoditi na več ustanovah (šole, Ljudska univerza), a jih tudi tam ne uporabljajo.

Odzivi otrok in staršev so izjemni. Izkazalo se je, da krepko podcenjujemo sposobnosti otrok. S pravilnim pristopom in primerno motivacijo, ki jo opisani programi vsekakor ponujajo, in seveda z nekaj nadzora in pomoči, nas bodo zagotovo s svojimi izdelki prijetno presenetili.

Poletna šola je pokazala, da so izbrani programi odlični pripomočki za učenje programiranja, saj so otroci navdušeni, njihovi projekti pa so potrditev, da so osnovnošolci željni in zmožni programirati. Potrebujejo le priložnost.

Literatura

- [1] Založba Rokus Klett d.o.o., „Dežela Lilibi,“ [Elektronski]. Available: <http://www.lilibi.si>. [Dostop 15. julij 2014].
- [2] Spletna založba, Zvonka Kos s.p., „Moja matematika,“ [Elektronski]. Available: <http://www.moja-matematika.si>. [Dostop 15. julij 2014].
- [3] „Nauk.si, Napredne učne kocke,“ [Elektronski]. Available: <http://www.nauk.si>. [Dostop 14. julij 2014].
- [4] „E-um,“ [Elektronski]. Available: <http://www.e-um.si>. [Dostop 14. julij 2014].
- [5] „Aktivna matematika,“ [Elektronski]. Available: <http://am.fmf.uni-lj.si/>. [Dostop 14. julij 2014].
- [6] „E-va,“ [Elektronski]. Available: <http://www.fiz.e-va.si/>. [Dostop 14. julij 2014].
- [7] „Portal izobraževalnih iger,“ [Elektronski]. Available: <http://hrast.pef.uni-lj.si/igre/>. [Dostop 10. avgust 2014].
- [8] A. P. Škraba. [Elektronski]. Available: <http://astra.si/>. [Dostop 10. avgust 2014].
- [9] „Theguardian,“ [Elektronski]. Available: <http://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick>. [Dostop 15. avgust 2014].
- [10] „CoderDojo,“ [Elektronski]. Available: <http://ljubljana.coderdojo.si/>. [Dostop 15. avgust 2014].
- [11] „CoderDojo,“ [Elektronski]. Available: <http://coderdojo.com>. [Dostop 16. avgust 2014].

- [12] „Code.org,“ [Elektronski]. Available: <http://code.org/>. [Dostop 17. avgust 2014].
- [13] S. Lokar, „www.dnevnik.si,“ [Elektronski]. Available: <http://www.dnevnik.si/posel/zaposl/osnovno-znanje-programiranja-lahko-izboljsa-nase-delo-in-poveca-zaposljivost>. [Dostop 16. avgust 2014].
- [14] „Daisy The Dinosaur,“ Hopscotch Tehnologies, [Elektronski]. Available: <https://itunes.apple.com/us/app/daisy-the-dinosaur/id490514278?mt=8>. [Dostop 25. julij 2014].
- [15] „Hopscotch,“ [Elektronski]. Available: <https://itunes.apple.com/us/app/hopscotch-coding-for-kids/id617098629?mt=8&ign-mpt=uo%3D4>. [Dostop 25. julij 2014].
- [16] „Dynamic ART,“ [Elektronski]. Available: <https://itunes.apple.com/us/app/dynamic-art-graphical-programming/id592737092?mt=8>. [Dostop 29. julij 2014].
- [17] „Kodu,“ [Elektronski]. Available: <http://research.microsoft.com/en-us/projects/kodu/>. [Dostop 29. julij 2014].
- [18] „Scratch,“ [Elektronski]. Available: <http://scratch.mit.edu/>. [Dostop 4. julij 2014].
- [19] „Alice,“ [Elektronski]. Available: <http://www.alice.org/index.php>. [Dostop 20. julij 2014].
- [20] „Move the Turtle,“ [Elektronski]. Available: <http://movetheturtle.com/>. [Dostop 22. julij 2014].
- [21] „MIT App Inventor,“ [Elektronski]. Available: <http://appinventor.mit.edu/explore/>. [Dostop 17. julij 2014].
- [22] „Kogics,“ [Elektronski]. Available: <http://www.kogics.net/kojo>. [Dostop 17. julij 2014].
- [23] „Greenfoot,“ [Elektronski]. Available: <http://www.greenfoot.org/door>. [Dostop 17. julij 2014].

- [24] „Tynker,“ [Elektronski]. Available: <http://www.tynker.com/>. [Dostop 23. julij 2014].
- [25] „LEGO Education,“ [Elektronski]. Available: <https://education.lego.com/en-us/lesi/elementary/wedo>. [Dostop 21. julij 2014].
- [26] „LEGO Mindsrorms,“ [Elektronski]. Available: <http://www.lego.com/en-us/mindstorms/?domainredir=mindstorms.lego.com>. [Dostop 22. julij 2014].
- [27] „Lightbot,“ [Elektronski]. Available: <http://light-bot.com/hocflash.html>. [Dostop 22. julij 2014].
- [28] „Cargo-Bot,“ [Elektronski]. Available: <https://itunes.apple.com/us/app/cargo-bot/id519690804?mt=8>. [Dostop 2. avgust 2014].
- [29] „Kodable,“ [Elektronski]. Available: <http://www.kodable.com/>. [Dostop 22. julij 2014].
- [30] S. Lajovic, Scratch, Ljubljana: Pasadena, 2011.

Priloga

Projekti udeležencev poletne šole programiranja

PREPOZNAJ BARVE (Rok, 13 let)

Smo v diskoteki. Na stropu je 7 oštevilčenih barvnih luči. Figura na sredini ekrana (5x) pove naključno izbrano barvo, mi pa moramo odtipkati številko te barve. Če smo odgovorili pravilno, nas figura zvočno pohvali in luč zasveti. Če se zmotimo, se figura prime za glavo, nam zaželi več sreče prihodnjič in igre je konec.

PUŠČAVA (Maria, 10 let)

Mucek v puščavi lovi papagaje. Modri, rumeni in zeleni prinašajo točke (na kar nas opozori tudi zvok), rdeči jih odnašajo. Ob živahni glasbi in hitrih papagajih mora mucek zbrati 7 točk, da je igre konec.

SNEŽAK (Ema, 10 let)

Igra ima 2 stopnji. Na prvi nam snežak postavi 10 računov seštevanja in nam za vsak pravilen odgovor podari točko, na drugi pa nas riba na enak način preizkuša v odštevanju. Igra je opremljena z navodili, glasbo, o pravilnosti rezultatov pa nas snežak in riba opominjata v oblăčkih.

PREŽENIMO OBLAKE (Lenart, 10 let)

Na nebu so trije oblaki. Pregnali jih bomo z znanjem poštevanke: vsak pravilen odgovor bo pregnal en oblak. Ko na nebu ne bo več nobenega oblaka, bo posijalo sonce, mucek pa bo

odšel skozi vrata. Igra je opremljena z navodili, mucek pa se z nami pogovarja preko ekrana in zvočnikov.

MRAVLJE IN SADJE (Mojca, 10 let)

Mravlja pobira banane, vendar mora biti pazljiva: če se dotakne rdečih banan, se igra konča. Ko mravlja nabere 10 šopov banan, se na mravljišču prikažejo vrata in mravlja lahko gre domov.

ZIMA (Jošt in Tim, 10 let)

Sneži. Na ekranu se pojavi število snežink, ki ji jih mora snežak pobarvati modro. Če po pomoti pobarva kakšno preveč, jo lahko spremeni nazaj v prvotno, belo barvo. Ko menimo, da je snežak pobarval ustrezno število snežink, z miško pritismo na tablico Naprej. Snežak odide v svoj iglu, med potjo pa nam pove, če smo nalogo pravilno rešili.